

NEW SNRANAL and BATANAL.

John Gipson

Last Revision:
January 26, 1999

I recently completed writing a new version of snranal. This program has a long history going back to Clara Kuehn. It is fair to say that about the only thing new snranal has in common with the old version is the name. A fundamental difference between my version of snranal and previous versions is that my version is station oriented, as opposed to baseline oriented. Some of the features of the new snranal include:

1. Command line driven.
2. Processes both S and X band data.
3. Comparison of predicted and measured station and source performance.
4. Graphics output dumped to the screen or to the printer, or not dumped at all.
5. Histograms of measured, achieved, and corrected SNRs and sigmas by station.
6. Time history of SNRs by station.

CALLING SNRANAL

Snranal is command line driven. If you just type snranal without any arguments, it will remind of the expected syntax:

```
snranal -sked filename -database name [-plot (screen, print, none)]  
      [-outdir output_directory] [-outfile filename] [-histogram yes,no]  
      [-observations yes,no]
```

You must always specify the sked file and the database name. The remaining flags, given in [...], are optional. You don't have to give to type all of the characters of the flags: only enough to distinguish them. For example, -sked can be abbreviated to -s, but -outdir can only be abbreviated to -outd. Flags are case insensitive, as is the database name. Complete or relative pathnames are supported. The database name does not need the \$ character at the beginning, although putting this on should not hurt.

If histogram is turned on snranal will produce histograms of measured and expected SNR. If observations is turned on snranal will produce plots of the SNR by observation for different stations.

WHAT SNRANAL DOES IN SEQUENTIAL ORDER.

1. The first thing snranal does is to run sked to generate a summary of the expected schedule. This generates a file of the form:
 databasename.lst

2. Following this, snranal runs sumry on the S-band and X-band databases to find out what actually happened. It generates files of the form:

databasenameS.sum
databasenameX.sum

3. Following this, snranal goes through these three files, matching up the observations. For each observation it extracts the stations, sources, predicted and measured SNR at each band, and predicted and actual integration time.

4. The model it uses for the station and source behavior is that during each experiment, the stations are a constant multiplicative factor weaker (or stronger) than predicted and so are the sources. Therefore, the measured SNRs are related to the predicted SNRs by:

$$\text{SNRmeas} = \text{SNRpred} * \text{Stat1_fact} * \text{Stat2_fact} * \text{Source_fact}$$

where the last three terms on the right hand side are correction factors. Snranal takes the log of this equation to obtain:

$$\log(\text{SNRmeas}) = \log(\text{SNRpred}) + \log(\text{Stat1_fact}) + \log(\text{Stat2_fact}) + \log(\text{Source_fact})$$

This equation can be solved in a least squares sense for the $\log(\text{Stat_fact})$ and $\log(\text{Source_fact})$. Snranal does this, assuming that "on average" all sources and all stations behave as expected. This last assumption is necessary because if all the sources got X times as strong, and all the stations \sqrt{X} as weak, the measured SNR would not change. You need some constraint to fix this "X" factor. Snranal imposes the constraint that on average the adjustments are 0:

$$\text{Sum } \log(\text{stat_fact}) - 2 * \text{Sum } \log(\text{src_fact}) = 0.$$

Snranal imposes the additional constraint that any source that is observed less than 5 times has it's nominal strength.

5. Summary Output.

Snranal produces a summary output of expected versus measured performance for all sources and stations. Any sources or stations that differ by more than 30% from their expected value are flagged. This summary is put in the output file specified on the command line. If no output file is specified, the output goes into the file:

databasename.snrsum

6. SNR histograms.

Snranal bins the S- and X-band into bins which are 5 wide. It also calculates the cumulative distribution of SNR, e.g., what percentage of points has an SNR below 35. These are written out to files on a station by station base. These have filenames of the form:

databasename.STAT.hst

If requested on the command line, these files are printed to the laser printer, or plotted on the screen.

7. Sigma histograms.

Snranal calculates the S- and X-band sigmas based on the SNR and the RMS spanned bandwidth. It bins these in these which are 5 ps wide. It also calculates the cumulative distribution of SNR, e.g., what percentage of points has a sigma below 35ps. These are written out to files on a station by station base. These have filenames of the form:

databasename.STAT.sig

If requested on the command line, these files are printed to the laser printer, or plotted on the screen.

8. SNR by time and source.

SNR creates files which contain each observation a station is in by time and source. These have filenames of the form:

databasename.STAT.obs.

BATANAL

Batanal is unix shell script which automates the use of snranal.

Batanal starts by calling excat. Once the user has picked one or more experiments using the VU command, and exited, batanal creates a file

/tmp/dosnr.xxx

where xxx are your user initials. This contains a sequence of commands to call snranal and do clean up. Batanal executes this script. The summary files are put on /data1/snrcat. Graphical output is printed, along with the summary files.

Batanal is a pretty simple script. If you want it to do something else, feel free to make copies and modify them. For example, you might want to put the output on tmp. This could be done by including

-outdir /tmp

in the command line. If you are Dave Shaffer, you might want to turn plotting off. This can be done by including

-plots none

on the command line.

BUGS

Send all bug reports, feature requests to jmg@gemini.gsfc.nasa.gov.

WHERE THEY LIVE

Currently, the executables for both snranal and batanal live on aquila in /mk3/bin. The source code lives on

/data18/mk3/src/solve/snranal.